AMENDMENTS TO THE SPECIFICATION:

Page 1, please delete the old title and substitute therefore as follows:

-- DATA PROCESSOR HAVING UNIFIED MEMORY ARCHITECTURE USING REGISTER TO OPTIMIZE MEMORY ACCESS —

Page 1, following the title, please insert the following paragraph:

-- **Cross-Reference to Related Application**

This application is a continuation of application Serial No. 09/991,705 filed on November 26, 2001, which is a continuation of application Serial No. 08/942,689 filed on September 29, 1997, now U.S. Patent No. 6,333,745. The contents of application Serial Nos. 09/991,705 and 08/942,689 are hereby incorporated herein by reference in their entirety.

Paragraph beginning at page 1, line 9, has been amended as follows:

[~~As an art of a~~] A conventional data processor using [~~the~~] unified memory architecture[~~, the art~~] is described in Japanese Patent Application Laid-Open 4-84192 [~~is known~~].

Paragraph beginning at page 1, line 12, has been amended as follows:

In this [~~art~~] processor, a memory for buffering data to be transferred between a CPU and a main memory once is provided so as to prevent the main memory from being occupied by access by the CPU, and a display controller provided for

controlling display on a display unit reserves a period for reading data from the main memory for display.

Paragraph beginning at page 1, line 18, has been amended as follows:

As an [~~art~~] example of a conventional data processor using [~~the~~] unified memory architecture, an apparatus which is structured so as to use a part of a memory as a Z buffer and a texture memory necessary for three-dimensional graphics in addition to a frame memory and to allow a rendering processor or a geometric processor for processing three-dimensional graphics to access it is also known.

Paragraph beginning at page 1, line 25, has been amended as follows:

Furthermore, [~~as~~] a processor for processing graphics at high speed[~~, there is an example~~] (Quoted Example 1) is described in "Three-dimensional CG Drawing LSI - 300000 Polygons/Second Realized by Personal Computer - (Nikkei Electronics, No. 640, July 17, 1995, pp 109 - 120)". This processor is provided with three kinds of memories, such as a texture memory, a frame buffer memory[~~,~~] and a local memory, as dedicated processor memories. This architecture is advantageous in respect of improvement of the performance, though it is not suited to a compact and inexpensive apparatus, such as an individual portable device, because a plurality of memories are necessary.

Paragraph beginning at page 2, line 9, has been amended as follows:

On the other hand, an example [~~that~~] in which graphic information 10 is

unified and the number of memories can be reduced compared with Quoted Example 1 is disclosed in Japanese Patent Application Laid-Open 5-257793 (Quoted Example 2). This graphic system has a CPU program, texture data, and a frame buffer which are unified in a main memory of a CPU.

Paragraph beginning at page 2, line 15, has been amended as follows:

According to each of the aforementioned [~~prior arts~~] processor systems, there is a problem [~~imposed~~] in that access to the main memory by the CPU is forced to wait due to access to the main memory which is being executed by the display controller or the rendering processor, and so the processing performance of the whole processor is degraded. Furthermore, the period for which the access to the main memory by the CPU is forced to wait depends on the access executed by the display controller or the rendering processor, so that the period for which the access to the main memory by the CPU is forced to wait cannot be confirmed beforehand. As a result, [~~according to the prior arts~~] in these systems, it is impossible to guarantee the degradation of the processing performance of the whole processor by less than a fixed limit.

Paragraph beginning at page 3, line 1, has been amended as follows:

Furthermore, in the aforementioned [~~prior arts, it is~~] processor systems, [~~realized on~~] the assumption has been made that the data processor has a memory system which is sufficiently quick so that the access capacity of the memory is several hundreds MB/s and the display data reading time can be reserved sufficiently. This assumption requires an expensive memory system and disturbs

miniaturization and cost reduction.

Paragraph beginning at page 4, line 20, has been amended as follows:

An object of the present invention is to provide a data processor using [~~the~~] unified memory architecture for reducing degradation of the processing performance of the whole processor.

Paragraph beginning at page 4, line 27, has been amended as follows:

Another object of the present invention is to provide a data processor, when a graphics memory to be accessed from both a CPU and a graphics processor is used, for optimizing the continuous time for one display access depending on the cache system of the CPU, for increasing the access efficiency of the memory, and for realizing high speed graphics display and the graphics processor.

Paragraph beginning at page 5, line 3, has been amended as follows:

To accomplish the above first object, the present invention [~~is~~] provides a data processor which has, for example, a CPU, a main memory, and a display controller for controlling display on a display unit and uses a part of the storage area of the main memory as a frame buffer for storing display data which is read by the display controller and displayed on the display unit, wherein the data processor has a memory controller for connecting to a memory bus connected to the main memory, a CPU bus connected to the CPU, and a local bus connected to the display controller, relaying the memory access sequence of the CPU to the main memory between the CPU bus and the memory bus, and relaying the memory access sequence of the

display controller to the main memory between the local bus and the memory bus, and the memory controller gives priority to relaying of the memory access sequence between the CPU bus and the memory bus over relaying of the memory access sequence between the local bus and the memory bus, and even if a memory access by the CPU to the main memory occurs at a maximum frequency restricted by the throughput of the CPU bus, the throughput of the memory bus is set to a value larger than the throughput of the CPU bus so that the memory access frequency of the display controller to the main memory which is necessary for display on the display unit is reserved.

Paragraph beginning at page 6, line 3, has been amended as follows:

By use of such a constitution, when relaying of the memory access sequence between the CPU bus and the memory bus is given priority over relaying of the memory access sequence between the local bus and the memory bus, it can be guaranteed to prevent [~~the~~] performance degradation due to waiting of [~~access by~~] the CPU for access to the main memory and to allow the display controller to effect a display without trouble by setting the throughputs of the CPU bus and memory bus as mentioned above.

Paragraph beginning at page 6, line 9, has been amended as follows:

Furthermore, to accomplish the above first object, the present invention [~~is~~] provides a data processor which has, for example, a CPU, a main memory, and a display controller for controlling display on a display unit and uses a part of the storage area of the main memory as a frame buffer for storing display data which is

read by the display controller and displayed on the display unit, wherein the data processor has a memory controller for connecting to a memory bus connected to the main memory, a CPU bus connected to the CPU, and a local bus connected to the display controller, relaying the memory access sequence of the CPU to the main memory between the CPU bus and the memory bus, and relaying the memory access sequence of the display controller to the main memory between the local bus and the memory bus and the memory controller has a means for freezing execution of the memory access sequence of the display controller to the main memory when the memory access sequence of the CPU to the main memory starts on the CPU bus during relaying of the memory access sequence of the display controller to the memory bus, releasing the freezing of execution of the memory access sequence of the display controller to the main memory after the memory access sequence of the CPU to the main memory ends, and executing the unexecuted portion of the memory access sequence.

Paragraph beginning at page 7, line 7, has been amended as follows:

By use of such a constitution, when access by the CPU to the main memory occurs, by freezing the access by the display controller, waiting [~~of access~~] by the CPU <u>for access</u> to the main memory can be guaranteed less than the cycle necessary for the freezing process and by releasing the freezing of the stopped access by the display controller to the main memory, the access can be restarted from the point of time when the freezing is released. Therefore, the degradation of the access efficiency of the display controller to the main memory can be reduced.

Paragraph beginning at page 7, line 25, has been amended as follows:

To accomplish the above second object, the data processor of the present invention has a CPU for generating drawing procedure information (drawing command) comprising the kind of graphic diagrams to be displayed and top parameters, a memory for storing the aforementioned drawing procedure written (write access) from the CPU and drawing data (bit map information) outputted to a display unit, and a graphics processor for executing drawing access to the drawing procedure information, storing drawing data in the memory, and further executing display and reading (display access) for outputting the drawing data to the display unit, and furthermore the graphics processor changes the display and reading timing for the memory according to the cache system of the cache memory for transferring data to the memory provided in the CPU.

Paragraph beginning at page 8, line 4, has been amended as follows:

The graphics processor is given cache system information indicating whether the cache system is a system for transferring [~~data of~~] a plurality of words continuously to the memory or a system for transferring [~~data for~~] each word individually from the CPU, and the continuous time of one display and reading in the former case is made shorter than that in the latter case.

Paragraph beginning at page 8, line 13, has been amended as follows:

To accomplish the above object, the graphics processor of the present invention executes drawing access for generating bit map information and display access for displaying display data on a display unit to a graphics memory having a

storage area [~~of~~] for drawing procedure information (drawing command) comprising the kind of graphic diagrams to be displayed and top parameters and an area for storing the bit map information to be outputted to the display unit and furthermore transfer system information indicating whether the write access of the drawing procedure information to the graphics memory is a system for transferring data of a plurality of words at continuous addresses or a system for transferring data for each word is set and the continuous time of one display access in the former case is made shorter than that in the latter case.

Paragraph beginning at page 8, line 26, has been amended as follows:

Furthermore, the graphics processor has a drawing unit for executing [~~the~~] a drawing access, a display controller for executing [~~the~~] a display access, an interface means for receiving data transferred from the CPU and executing the write access, and a memory controller means for receiving those memory access requests and controlling them so as to give priority to the display access, and the transfer system information is set by the cache system of the CPU for transferring the drawing procedure information, and when the cache system is a copy back system, the continuous time of one display access is made shorter than that in the write through system.

Paragraph beginning at page 9, line 10, has been amended as follows:

Furthermore, the graphics processor has a display buffer for temporarily storing the bit map information [~~temporarily~~] which is read from the graphics memory by the display access and outputting it [~~in~~] at the timing of the display unit, and the

number of continuous words by one display access mentioned above is set as a maximum value of the number of held data of the display buffer, and the timing of request issue of the display access is set as a threshold value less than the number of held data, and the maximum value and threshold value vary with the transfer system information.

Paragraph beginning at page 13, line 18, has been amended as follows:

Fig. 43 is [~~an illustration for~~] a diagram showing the terminal functions of a graphics processor.

Paragraph beginning at page 13, line 20, has been amended as follows:

Fig. 44 is [~~an illustration for~~] a diagram showing the drawing commands of a graphics processor.

Paragraph beginning at page 13, line 22, has been amended as follows:

Fig. 45 is [~~an illustration for~~] a diagram showing the register functions of a graphics processor.

Paragraph beginning at page 13, line 24, has been amended as follows:

Fig. 46 is [~~an illustration for~~] a diagram of the functions of CPU FIFO.

Paragraph beginning at page 13, line 26, has been amended as follows:

Fig. 47 is [~~an illustration for~~] a diagram of the functions of a drawing cache (3).

Paragraph beginning at page 13, line 28, has been amended as follows:

Fig. 48 is [~~an illustration for~~] a diagram showing address mapping of an CPU.

Paragraph beginning at page 14, line 9, has been amended as follows:

[~~The~~] Various embodiments of the data processor of the present invention will be explained hereunder.

Paragraph beginning at page 14, line 11, has been amended as follows:

The [~~shown~~] data processor described herein can be applied to an electronic computer, such as a personal computer, and to a game machine.

Paragraph beginning at page 14, line 14, has been deleted in its entirety as follows:

[~~Firstly, the first embodiment will be explained.~~]

Paragraph beginning at page 14, line 15, has been amended as follows:

The constitution of the data processor [~~of this~~] representing a first embodiment is shown in Fig. 1.

Paragraph beginning at page 14, line 17, has been amended as follows:

In Fig. 1, [~~numeral 100 indicates a CPU, 115 a CRT, 120~~] the data processor includes a CPU 100, a CRT 115, a memory[~~, 140~~] 120, a memory controller[~~, 150~~] 140, a CPU interface circuit[~~, 151~~] 150, an internal bus arbitration circuit[~~, 152~~] 151, a memory interface circuit[~~, 155~~] 152, a DAC[~~, 156~~] 155, a display controller[~~, 157~~]

156, a rendering processor[, ~~170~~] 157, a memory bus control circuit[, ~~180~~] 170, a holding buffer[, ~~101~~] 180, a bus bridge 101, [~~102~~] a system bus[, ~~103~~] 102, an I/O device[, ~~104~~] 103, a hard disk drive 104, [~~105~~] a CD-ROM drive[, ~~and 106~~] 105, an input device 106, such as a keyboard, mouse, input tablet, or joystick.

Paragraph beginning at page 14, line 28, has been amended as follows:

A CPU bus 131, a memory bus 135, an internal bus 153, and the system bus 102 comprise an address bus and a data bus, respectively. The storage space of the memory 120 is divided into a main memory 121, which is a storage space into which a program and data necessary for the CPU 100 to operate are loaded from the CD-ROM drive 105 and the hard disk drive 104 via the bus bridge 101 and which is used by the CPU 100 as a working area, and a frame buffer 122, which is a storage space for storing display data to be displayed on the CRT 115.

Paragraph beginning at page 15, line 9, has been amended as follows:

By use of such a constitution, the CPU 100 executes data processing according to an input from the input device 106 by accessing the main memory 121 and stores display data to be displayed on the CRT 115 in the frame buffer 122. The rendering processor 157 performs [~~the~~] a rendering process by accessing texture data stored in the main memory 121 and stores display data to be displayed on the CRT 115 in the frame buffer 122. The display data stored in the frame buffer 122 is read by the display controller 156 and displayed on the CRT 115.

Paragraph beginning at pag 15, line 18, has been amended as follows:

[~~The~~] An outline of the operation of this data processor will be explained hereunder.

Paragraph beginning at page 15, line 25, has been amended as follows:

(1) Firstly, consideration will be given to an operation when the CPU accesses the main memory 121 [~~when~~] and the memory bus 135 executes no valid memory access.

Paragraph beginning at page 18, line 7, has been amended as follows:

(2) [~~An~~] Consideration will now be given to an operation when the display controller 156 or the rendering processor 157 accesses the frame buffer 122 [~~when~~] and the memory bus 135 executes no valid memory access (the memory bus 135 is not used for memory access).

Paragraph beginning at page 19, line 2, has amended as follows:

When the corresponding data is read from the frame buffer 122 in the memory 120, it is transferred to the display controller 156 via the memory bus 135, the memory interface circuit 152, and the internal bus 153. When the display controller 156 finishes the access, it sets reqA to 0 and [~~notifies~~] indicates an access end to the internal bus arbitration circuit 151, and the internal bus arbitration circuit 151 sets ackA to 0.

Paragraph beginning at page 19, line 12, has been amended as follows:

On the other hand, when the rendering processor 157 makes a write access

to the memory 120, it sets and outputs reqB to the internal bus arbitration circuit 151 to 1 and outputs a write request comprising a write command, a write address, and the number of request data and write data to the internal bus 153 when the internal bus arbitration circuit 151 sets ackB to 1. When there is no access by the CPU 100, the switching signal 175 is set at 0, so that the write request outputted to the internal bus 153 is outputted to the memory 120 via the memory interface circuit 152 and the memory bus 135. When the rendering processor 157 finishes the access, it sets reqB to 0 and [~~notifies~~] indicates an access end to the internal bus arbitration circuit 151, and the internal bus arbitration circuit 151 sets ackB to 0.

Paragraph beginning at page 20, line 4, has been amended as follows:

(3) An operation when an access request is outputted from the internal bus 153 [~~when~~] and the memory bus 135 is executing access from the CPU 100 will be considered.

Paragraph beginning at page 20, line 11, has been amended as follows:

(4) An operation when an access request is outputted from the CPU 100 [~~when~~] and the memory bus 135 is executing access from the internal bus 153 will be considered.

Paragraph beginning at page 20, line 18, has been amended as follows:

Firstly, an operation when an access request (read request or write request) is outputted from the CPU 100 [~~when~~] and read access by the rendering processor 157 is in execution will be explained. When an access request is outputted from the

CPU 100, the CPU bus control signal 132 is inputted to the memory bus control circuit 170. The memory bus control circuit 170 sets the switching signal 175 to 1 in the same timing as that of the operation when the memory bus 135 executes no valid memory access and allows execution of access from the CPU 100. If the access of the rendering processor 157 which is in execution at present does not end before the switching signal 175 is set to 1, the memory bus control circuit 170 sets a stop signal 173 to 1 before setting the switching signal 175 to 1 and [~~notifies~~] indicates a halting of the access [~~in execution~~] being executed at present to the rendering processor 157.

Paragraph beginning at page 21, line 25, has been amended as follows:

When the data transferred from the memory 120 is stored in the holding buffer 180, the memory bus control circuit 170 suspends the current access [~~in execution~~] and outputs a control signal for precharging the memory 120 to the memory 120 as the internal bus control signal 154 via the internal bus 153, the memory interface circuit 152, and the memory bus 135. Hereafter, the memory bus control circuit 170 sets the switching signal 175 to 1 and allows execution of access of the CPU 100. When the CPU 100 executes access, the memory bus control circuit 170 sets the switching signal 175 to 0 and allows execution of access from the internal bus 153. To restart the suspended access of the rendering processor 157 in synchronization with the timing [~~that~~] at which the switching signal 175 changes from 1 to 0, the holding buffer 180 outputs the held read request to the internal bus 153. To output the data which is read from the memory 120 after the stop signal 173 is set to 1 and held in the holding buffer 180 to the internal bus 153

before setting the stop signal 173 to 0, the memory bus control circuit 170 outputs a pop signal. The holding buffer 180 outputs the held read data in synchronization with the pop signal.

Paragraph beginning at page 24, line 19, has been amended as follows:

In the case of write access, an access request (Active command) including a write command and the number of request data (burst length) is given to the memory together with the row address of the memory in the first cycle, and the write command and the column address of target data and the data are given at the same time sequentially in each cycle after the cycle is spread by the write RAS-CAS latency cycle, and the precharge command is given in the last cycle for the next access sequence. In this case, the write RAS-CAS latency indicates the number of cycles until the first write command and column address can be given after the Active command and the row address are given, [~~and~~] the write CAS latency indicates the number of cycles (generally 0) until the data at a column address is given after the write command and the column address are given, and the precharge latency indicates the number of cycles until the Active command including the next write command can be given after the precharge command is given at the preceding step.

Paragraph beginning at page 24, line 27, has been amended as follows:

However, as mentioned above, generally, there exists no CAS latency for writing and the RAS-CAS latency generally takes the same value for both reading and writing. Therefore, hereunder, in the case [~~that just~~] <u>where only</u> RAS-CAS

latency is described, it indicates an AS-CAS latency value common to reading and writing and in the case [~~that just~~] where only CAS latency is described, it indicates a CAS latency value for reading.

Paragraph beginning at page 25, line 8, has been amended as follows:

Firstly, the constitution of the CPU interface circuit 150 [~~is shown in~~] will be described with reference to Fig. 2.

Paragraph beginning at page 25, line 19, has been amended as follows:

Next, the constitution of the memory interface circuit 152 [~~is shown in~~] will be described with reference to Fig. 3.

Paragraph beginning at page 26, line 8, has been amended as follows:

Next, the constitution of the memory bus control circuit 170 [~~is shown in~~] will be described with reference to Fig. 4.

Paragraph beginning at page 26, line 9, has been amended as follows:

In Fig. 4, [~~numerals 410 and 411 indicate decoders, 420~~] the memory bus control circuit includes decoders 410 and 411, a CPU access buffer [~~, 421~~] 420, a CPU bus access length calculation circuit [~~, 422~~] 421, an internal bus access counter 422, [~~423~~] an internal bus access length calculation circuit [~~, 430~~] 423, a stop decision circuit [~~, 440~~] 430, a switching signal generation circuit [~~, 441~~] 440, a stop processing circuit 441, and [~~442~~] a restart processing circuit 442.

Paragraph beginning at pag 27, line 22, has been amended as follows:

The decoder 411 decodes the internal bus control signal 154 and outputs an internal bus access start signal which becomes 1 when an access request is issued from the internal bus 153, a command indicating whether the access request from the internal bus 153 is reading or writing, and a burst length indicating the number of read or write data. The internal bus access counter 422 resets the internal counter at the timing that it receives the internal bus access start signal, counts the number of cycles during execution of internal bus access, and outputs it as an internal access cycle. The internal bus access counter 422 further outputs the count value of the internal counter when the CPU bus access request signal becomes 1 as a stop request cycle. The internal bus access length calculation circuit 423 calculates the number of cycles required for access from the internal bus 153 from the command and burst length and outputs it as an internal access length. the stop decision circuit 430 outputs a switching start cycle indicating the timing for setting the switching signal 175 to 1, on the basis of the command outputted from the decoder 411, the stop request cycle[,] and the internal access length, a stop start cycle indicating the timing for setting the stop signal 173 to 1, and a data holding signal indicating whether there is read data to be transferred to the internal bus 153 from the memory 120 or not during a period from the time that the stop signal 173 becomes 1 to the time that the switching signal 175 becomes 1 when the access in execution from the internal bus 153 is a read access.

Paragraph beginning at page 28, line 2, has been amended as follows:

The switching signal generation circuit 440 controls the value of the switching

signal 175 on the basis of the CPU access length, internal access cycle, and switching start cycle. The stop processing circuit 441 outputs the stop signal 173 on the basis of the command outputted from the decoder 411, the CPU access length, the internal access cycle, the stop start cycle, and a data holding signal. The stop processing circuit 441 also outputs a push signal when there is read data to be transferred to the internal bus 153 from the memory 120 during a period from the time that the stop signal 173 becomes 1 to the time that the switching signal 175 becomes 1 and outputs a signal for precharging the memory 120 to the internal bus 153 as a control signal when it is necessary to suspend the access from the internal bus 153 which is in execution at present. When [~~there is suspended~~] access from the internal bus 153 is suspended and the suspended access is a read access, the restart processing circuit 442 outputs a signal for requesting to output read data held in the holding buffer 180 to the internal bus 153 as a pop signal on the basis of the CPU access length, the internal access cycle, the switching signal 175, and the stop signal 173.

Paragraph beginning at page 28, line 27, has been amended as follows:

The constitution of the CPU access buffer 420 shown in Fig. 4 [is shown in] will be described with reference to Fig. 5.

Paragraph beginning at page 29, line 8, has been amended as follows:

[~~The constitution~~] An example of the CPU bus access length calculation circuit 421 shown in Fig. 4 [~~is shown in~~] will be described with reference to Fig. 6.

Paragraph beginning at page 29, line 22, has been amended as follows:

The constitution of the internal bus access counter 422 shown in Fig. 4 [~~is shown in~~] will be described with reference to Fig. 7.

Paragraph beginning at page 29, line 28, has been amended as follows:

In Fig. 7, the counter 710 is reset to 1 by the internal bus access start signal and then counts the number of execution cycles of access from the internal bus 153 and outputs it as an internal access cycle. At the timing [~~that~~] when the CPU bus access request signal is set to 1, the CPU bus access length calculation circuit 421 outputs the output from the counter 710 as a stop request cycle. In other cases, the CPU bus access length calculation circuit 421 outputs "1000" as a sufficiently large default value.

Paragraph beginning at page 30, line 7, has been amended as follows:

The constitution of the internal bus access length calculation circuit 423 shown in Fig. 4 [~~is shown in~~] will be described with reference to Fig. 8.

Paragraph beginning at page 30, line 20, has been amended as follows:

The constitution of the stop decision circuit 430 shown in Fig. 4 [is shown in] will be described with reference to Fig. 9.

Paragraph beginning at page 30, line 21, has been amended as follows:

In Fig. 9, [~~numeral 910 indicates~~] the stop decision circuit 430 includes a transfer time register [~~, 911~~] 910, a memory characteristic register [~~, 920~~] 911, a read

stop decision circuit 920, and [~~, and 930~~] a write stop decision circuit 930.

Paragraph beginning at page 31, line 24, has been amended as follows:

The constitution of the read stop decision circuit 920 shown in Fig. 9 [~~is shown in~~] will be described with reference to Fig. 10.

Paragraph beginning at page 31, line 25, has been amended as follows:

In Fig. 10, [~~numerals 1010 and 1015 indicate~~] the read stop decision circuit 920 includes comparators 1010 and 1015, [~~1011~~] an adder 1011, [~~1012 and 1014 subtracters~~] subtractors 1012 and 1014, [~~1013~~] a counter 1013, [~~1016~~] a selector 1016, [~~, and 1017~~] an AND circuit 1017.

Paragraph beginning at page 32, line 21, has been amended as follows:

The constitution of the write stop decision circuit 930 shown in Fig. 9 [~~is shown in~~] will be described with reference to Fig. 11.

Paragraph beginning at page 32, line 22, has been amended as follows:

In Fig. 11, [~~numeral 1110 indicates a comparator, 1111~~] the write stop decision circuit 930 includes a comparator 1110, an adder [~~, 1112~~] 1111, a [~~subtracter, and 1113~~] subtractor 1112, and a selector 1113.

Paragraph beginning at page 33, line 6, has been amended as follows:

Next, [~~a constitution~~] an example of the switching signal generation circuit 440 of the memory bus control circuit 170 shown in Fig. 4 will be [~~shown~~] described with

reference to Fig. 12.

Paragraph beginning at page 33, line 9, has been amended as follows:

In Fig. 12, [~~numerals 1210 and 1212 indicate~~] the switching signal generation circuit 440 includes comparators [~~, 1211~~] 1210 and 1212, an adder [~~, and 1220~~] 1211, a switching resister 1220.

Paragraph beginning at page 33, line 22, has been amended as follows:

Next, the constitution of the stop processing circuit 441 shown in Fig. 4 [~~is shown in~~] will be described with reference to Fig. 13.

Paragraph beginning at page 33, line 23, has been amended as follows:

In Fig. 13, [~~numeral 1310 indicates~~] the stop processing circuit 441 includes a memory characteristic register [~~, 1320 and 1322~~] 1310, comparators [~~, 1321~~] 1320 and 1322, an adder [~~, 1323~~] 1321, a counter [~~, 1324~~] 1323, an FIFO buffer [~~, 1325~~] 1324, a selector [~~, 1330~~] 1325, a stop register [~~, and 1331~~] 1330, a precharger issue circuit 1331.

Paragraph beginning at page 34, line 15, has been amended as follows:

In Fig. 13, the memory characteristic register 1310 is a register holding a RAS-CAS latency, a read precharge latency, and a write precharger latency. The comparator 1320 outputs 1 when the internal access cycle is equal to the stop start cycle. The selector 1325 outputs the read or write precharger latency depending on the command outputted from the decoder 411. The adder 1321 outputs the sum of

the stop start cycle, CPU access length, RAS-CAS latency, and output value of the selector 1325. The comparator 1322 outputs 1 when the internal access cycle is equal to the sum of the adder 1321. The counter 1323 outputs 1 for the count shown by the output value of the selector 1325 at the timing that the output value of the comparator 1320 becomes 1. The FIFO buffer 1324 holds a data holding signal and outputs the held data holding signal as a push signal at the timing [~~that~~] when the output value of the counter 1323 becomes 1. The stop register 1330 is set to 1 when the output value of the comparator 1320 is I and reset to 0 when the output value of the comparator 1322 is 1. The value held by the stop register 1330 is outputted as the stop signal 173. The precharger issue circuit 1331 outputs a control signal for precharging the memory 120 to the internal bus 153 at the timing [~~that~~] when the output value of the comparator 1320 becomes 1.

Paragraph beginning at page 34, line 24, has been amended as follows:

[~~A constitution~~] An example of the restart processing circuit 442 shown in Fig. 4 [~~is shown in~~] will be described with reference to Fig. 14. In Fig. 14, [~~numeral 1410 indicates~~] the restart processing circuit 442 includes a memory characteristic register [~~, 1420~~] 1410, a latch [~~, 1421 and 1422~~] 1420, AND circuits 1421 and 1422, and [~~1423~~] a counter 1423.

Paragraph beginning at page 35, line 7, has been amended as follows:

In Fig. 14, the memory characteristic register 1410 is a register holding a RAS-CAS latency. The AND circuit 1421 outputs 1 at the timing that the switching signal 175 is switched from 1 to 0. The AND circuit 1422 outputs 1 when the output

value of the AND circuit 1421 is 1 and the stop signal 173 is 1. The counter 1423 outputs 1 as a pop signal for the RAS-CAS latency cycle at the timing [~~that~~] when the output value of the AND circuit 1422 becomes 1.

Paragraph beginning at page 35, line 9, has been amended as follows:

The memory bus control circuit 170 [~~is~~] has been explained above.

Paragraph beginning at page 35, line 11, has been amended as follows:

Next, the constitution of the holding buffer 180 shown in Fig. 1 [~~is shown in~~] will be described with reference to Fig. 15.

Paragraph beginning at page 35, line 12, has been amended as follows:

In Fig. 15, [~~numeral 1510 indicates~~] the holding buffer 180 includes a decoder [~~, 1520~~] 1510, a suspended access register [~~, 1530~~] 1520, a write permission buffer [~~, 1540~~] 1530, a data holding register [~~, 1550, 1553, and 1555~~] 1540, AND [~~circuits, and~~] circuits 1550, 1553, 1555 and latches 1551, 1552, and 1554 [~~latches~~].

Paragraph beginning at page 35, line 19, has been amended as follows:

In Fig. 15, the decoder 1510 decodes an access request read from the internal bus 153 and outputs a command indicating whichever the suspended access is, reading or writing, and the suspended address of the suspended access. The AND circuit 1550 outputs 1 at the timing that the switching signal 175 is switched from 1 to 0 and the stop signal 730 is 1. The suspended access register 1520 holds a command and the suspended address and outputs the held command

and suspended address to the internal bus 153 at the timing that the output value of the AND circuit 1550 becomes 1. The AND circuit 1553 outputs 1 at the timing [~~that~~] when the stop signal 173 is switched from 0 to 1. The AND circuit 1550 outputs 1 at the timing [~~that~~] when the switching signal 750 is switched from 0 to 1. The write permission buffer 1530 is set to 1 when the output value of the AND circuit 1553 is 1 and reset to 0 when the output value of the AND circuit 1555 is 1. The write permission buffer 1530 outputs the held value as a write permission signal. The data holding register 1540 reads data from the internal bus 153 while the write permission signal is 1 and holds a push signal as a valid flag at the same time. The data holding register 1540 also outputs the held data to the internal bus 153 in the reading order while a pop signal is 1. In this case, only when the valid flag is 1, the data holding register 1540 outputs valid data.

Paragraph beginning at page 36, line 13, has been amended as follows:

Each unit of the memory controller 140 [~~is~~] has been explained in detail above.

Paragraph beginning at page 36, line 17, has been amended as follows:

Details realized by the aforementioned operation of each unit among the operations of this data processor whose outline [~~is~~] has been described previously will be indicated below.

Paragraph beginning at page 36, line 18, has been amended as follows:

Fig. 16 shows a case [~~that~~] in which a read access request is issued from the

CPU bus 131 during execution of read access from the internal bus 153. Fig. 16 also shows 5 cycles of transfer time, 2 cycles of RAS-CAS latency, 3 cycles of CAS latency, and 3 cycles of read precharge latency. The operation timing T of the memory bus 135 is represented as one cycle.

Paragraph beginning at page 37, line 7, has been amended as follows:

In Fig. 16, read access from the internal bus 153 starts at T=I. At this timing, the read command and row address of read access are registered in the holding buffer 180. A read access request occurs from the CPU bus 131 at T=3. However, to execute the access from the CPU bus 131 in a transfer time of 5 cycles, it is necessary to switch the memory bus 135 to access from the CPU bus 131 at T=8 and to suspend the read access in execution. Therefore, the stop signal 173 becomes 1 at T=5. Although the switching signal 175 becomes 1 at T=8, read data is transferred from the memory 120 at the timing of T=6 and T=7 by the read command issued at the timing of T=3 and T=4, so that the push signal is set to 1 at T=6 and T=7 and data d0 and dI are held in the holding buffer 180. To suspend the access in execution and execute the access from the CPU bus 131 at T=8, a precharge command is outputted to the internal bus 153 from the memory bus control circuit 170 at T=5. The switching signal 175 is set to 1 at T=8 and the read access from the CPU bus 131 is executed until T=16. The switching signal 175 becomes 0 at T=17 and the command and row address held by the holding buffer 800 are outputted to the internal bus 153 at this timing. The stop signal 173 becomes 0 at T=19 and the pop signal becomes 1 in synchronization with it. The pop signal is kept at 1 between T=19 and T=21. However, since no valid data is

held, no data is outputted to the internal bus 153 at T=19. At T=20 and T=21, the data d0 and dl are outputted to the internal bus 530.

Paragraph beginning at page 37, line 25, has been amended as follows:

Next, an example [~~that~~] in which a read access request is issued from the CPU bus 131 during execution of write access from the internal bus 153 is shown in Fig. 17 as a time chart of 5 cycles of transfer time, 2 cycles of RAS-CAS latency, 0 cycle of CAS latency, and 2 cycles of write precharge latency. The time chart shown in Fig. 17 is represented on the basis of the operation timing T of the memory bus 135.

Paragraph beginning at page 38, line 26, has been amended as follows:

The first embodiment of the present invention [~~is~~] has been explained above.

Paragraph beginning at page 39, line 2, has been amended as follows:

In Fig. 18, numeral 210 indicates a read buffer, 1810 denotes a cache memory control circuit, and 1820 denotes a cache memory.

Paragraph beginning at page 40, line13, has been amended as follows:

As mentioned above, according to the first embodiment, the waiting time for access to the memory 120 by the CPU 100 can be guaranteed by the aforementioned transfer time value. Therefore, compared with the conventional method, the degradation of processing performance can be reduced. Access from the internal bus 153 is forcibly suspended when access from the CPU 100 occurs,

though the access sequence can be restarted from the suspended location after the access from the CPU 100 ends. Therefore, the reduction in access efficiency from the internal bus due to use of [~~the constitution of~~] this embodiment is extremely small.

Paragraph beginning at page 40, line 17, has been amended as follows:

The constitution of [~~the~~] a data processor [~~of the~~] representing a second embodiment is shown in Fig. 19.

Paragraph beginning at page 40, line 21, has been amended as follows:

As shown in the drawing, the constitution of the data processor of the second embodiment is almost the same as [~~the constitution~~] that of the data processor of the first embodiment, [~~though~~] and only the constitution of a memory interface circuit 1952 and [~~the~~] a modification [~~that~~] in which a cancel signal is sent from the memory interface circuit 1952 to a display controller 1956 and a rendering processor 1957 are different.

Paragraph beginning at page 41, line 13, has been amended as follows:

[~~A constitution~~] An example of the memory interface circuit 1952 shown in Fig. 19 [~~is shown in~~] will be described with reference to Fig. 20. In Fig. 20, numeral 2010 indicates an address monitor circuit and 2020 indicates a permitted address register. The other elements are the same as the elements to which the same numerals are assigned in the first embodiment.

Paragraph beginning at page 42, lin 5, has been am nded as follows:

[~~The~~] A third embodiment of the present invention will be explained hereunder.

Paragraph beginning at page 42, line 10, has been amended as follows:

In Fig. 21, numeral 3656 indicates a display controller, 3657 denotes a rendering processor, and 3670 denotes a memory bus control circuit. The other units are the same as the units to which the same numerals are assigned in Fig. 1. As shown in the drawing, the third embodiment has a constitution [~~that~~] in which the holding buffer 180 is omitted from the [~~constitution of the~~] first embodiment shown in Fig. 1.

Paragraph beginning at page 42, line 16, has been amended as follows:

Firstly, [~~the~~] an outline of the operation of the data processor of the third embodiment will be explained.

Paragraph beginning at page 42, line 20, has been amended as follows:

In the data processor, the operation when one of the CPU 100, the display controller 3656, and the rendering processor 3657 accesses the memory 120 [~~when~~] and the memory bus 135 executes no valid memory access is the same as the operation in the first embodiment. The operation when an access request is issued from the internal bus 153 during memory access by the CPU 100 is also the same as that of the first embodiment.

Paragraph beginning at page 43, line 1, has been amended as follows:

On the other hand, if an access request is issued from the CPU 100 when the memory bus 350 is executing access from the internal bus 153, the data processor operates as [~~shown~~] described below.

Paragraph beginning at page 43, line 3, has been amended as follows:

Even if access to the memory bus 135 from the internal bus 153 is [~~one~~] by a read request or [~~one~~] by a write request, the operation of the memory controller 140 which is a center of operation is not changed. Therefore, a case [~~that~~] in which an access request from the CPU 100 occurs when the rendering processor 3657 is executing a read access will be explained hereunder as an example.

Paragraph beginning at page 43, line 19, has been amended as follows:

When an access request is outputted from the CPU 100, the CPU bus control signal 132 is inputted to the memory bus control circuit 3670 in the same way as with the first embodiment. The memory bus control circuit 3670 sets the switching signal 175 to 1 and allows execution of access from the CPU 100 at the same timing as that when the memory bus 135 executes no valid memory access. When the memory access by the rendering processor 3657 does not end before the switching signal 175 becomes 1, the memory bus control circuit 3670 sets the stop signal 173 to 1 before setting the switching signal 175 to 1 and [~~notifies~~] indicates a halting of the access in execution at present to the rendering processor 3657. When the stop signal 173 becomes 1 during execution of the access, the rendering processor 3657 makes the data read up to halfway invalid and stops until the stop signal 173

becomes 0.

Paragraph beginning at page 43, line 27, has been amended as follows:

The memory bus control circuit 3670 stops the memory access of the rendering processor 3657 in this way and then sets the switching signal 175 to 1 and allows execution of an access by the CPU 100. When the access by the CPU 100 ends, the memory bus control circuit 3670 sets the switching signal 175 to 0 and allows execution of an access to the memory 120 from the internal bus 153. The memory bus control circuit 3670 sets the switching signal 175 to 0 and also sets the stop signal 173 to 0 at the same time. When the stop signal 173 becomes 0, the rendering processor 3657 which is stopped executes the access which [~~is~~] has been suspended due to changing of the stop signal 173 from 0 to 1 from the beginning of the side.

Paragraph beginning at page 45, line 8, has been amended as follows:

The constitution of the stop decision circuit 3730 shown in Fig. 22 [~~is shown in~~] will be described with references to Fig. 23.

Paragraph beginning at page 45, line 22, has been amended as follows:

The selectors 940 and 941 select an output from the read stop decision circuit 3820 or an output from the write stop decision circuit 930 according to a command outputted from the decoder 411 and [~~outputs~~] output it as a switching start cycle or a stop start cycle.

Paragraph beginning at page 45, line 28, has been amended as follows:

The constitution of the read stop decision circuit 3820 shown in Fig. 23 [~~is shown in~~] will be described with reference to Fig. 24.

Paragraph beginning at page 46, line 12, has been amended as follows:

Next, the constitution of the stop processing circuit 3741 shown in Fig. 22 [~~is shown in~~] will be described with reference to Fig. 25.

Paragraph beginning at page 46, line 26, has ben amended as follows:

An actual example of the operation whose outline [~~is~~] has been described before which is realized by the aforementioned constitution will be indicated hereunder.

Paragraph beginning at page 47, line 23, has been amended as follows:

The third embodiment of the present invention [~~is~~] has been explained above.

Paragraph beginning at page 47, line 25, has been amended as follows:

[~~The~~] A fourth embodiment of the present invention will be explained hereunder.

Paragraph beginning at page 48, line 2, has been amended as follows:

As shown in the drawing, the data processor of the fourth embodiment has a constitution [~~that~~] in which a compression and recovery circuit 2110 is added to the data processor of the first embodiment shown in Fig. 1.

Paragraph beginning at page 48, line 23, has been amended as follows:

As mentioned above, when the compression and recovery circuit 2110 for monitoring the address of access from the internal bus 153 and for selecting whether or not to execute the compression/expansion process depending on the address is installed, control such that, for example, the compression/expansion process is performed for access to the source data of the frame buffer 122 or the texture but the compression/expansion process is not performed for access to the display list of the rendering processor 157 is made possible. Therefore, control such that data in which it is necessary that data before compression and data which is expanded after compression coincide with each other like a display list is transferred as it is and data in which it is not always necessary that data before compression and data which is expanded after compression coincide with each other like pixel data in the frame buffer 122 is compressed and/or expanded and then transferred is made possible.

Paragraph beginning at page 49, line 22, has been amended as follows:

In the following explanation, a case [~~that~~] where compression for reducing the data amount of pixels is carried out by approximating the color of four pixels to the color of two ,25 pixels selected from the four pixels is used as an example. Therefore, in the following example, when compressed data is recovered, the original image before compression is not always recovered as it is.

Paragraph beginning at page 50, line 3, has been amended as follows:

In the drawing, [~~numeral 2210 indicates~~] the circuit 2110 includes an address conversion circuit [~~, 2220~~] 2210, a compression circuit [~~, 2230~~] 2220, a recovery circuit [~~, 3610~~] 2230, an address monitor circuit [~~, 3620~~] 3610, a compression execution address register [~~, and~~] 3620, and selectors 3630 and 3631 [~~selectors~~].

Paragraph beginning at page 51, line 1, has been amended as follows:

The constitution of the compression circuit 2220 shown in Fig. 28 [~~is shown in~~] will be described with reference to Fig. 29.

Paragraph beginning at page 51, line 3, has been amended as follows:

In Fig. 29, numeral 2310 indicates a primary color register, 2320 denotes a compression processing circuit, and 2330 denotes a compression register.

Paragraph beginning at page 51, line 8, has been amended as follows:

In the drawing, it is assumed that data before compression is data in pixel units (16 bits per pixel) and comprises fields of R (red, 5 bits), G (green, 6 bits), and B (blue, 5 bits). A case [~~that~~] where the display controller 156 reads and accesses this pixel data from the frame buffer 122 will be explained hereunder as an example.

Paragraph beginning at page 51, line 23, has been amended as follows:

The constitution of the compression processing circuit 2320 shown in Fig. 29 [~~is shown in~~] will be described with reference to Fig. 30.

Paragraph beginning at page 51, line 25, has been amended as follows:

In Fig. 30, numeral 2410 indicates a comparison data creation circuit, 2420 to 2425 and 2452 denote comparators, 2430 denotes an auxiliary color creation circuit, and 2450 and 2451 denote subtractors [~~subtracters~~].

Paragraph beginning at page 51, line 2, has been amended as follows:

The comparison data creation circuit 2410 creates and outputs comparison data for [~~deciding~~] determining the size of primary color data. Since data of each pixel comprises three different fields of R, G, and B, comparison data is data which is processed from the primary color data so as to prevent the relative sizes during comparison from deviation due to color. Comparison data for four primary color data is created and then primary color data corresponding to the maximum comparison data is selected by the comparators 2420 to 2422 and the selector controlled by the output of each comparator as a first color and primary color data corresponding to the minimum comparison data is selected by the comparators 2423 to 2425 and the selector controlled by the output of each comparator as a second color.

Paragraph beginning at page 52, line 18, has been amended as follows:

The auxiliary color creation circuit 2430 creates and outputs the difference between the first color and the second color as an auxiliary color 12 bits in length. The subtracters 2450 and 2451 and the comparator 2452 decide which color the primary color data is closer to, the first color or the second color. When it is closer to the first color, they output 1 as a selection number and when it is closer to the second color, they output 0 as a selection number. The selection number is outputted for each of the 4 primary color data stored in the primary color register

2310, so that 4 bits in total are outputted.

Paragraph beginning at page 52, line 26, has been amended as follows:

The constitution of the comparison data circuit 2410 shown in Fig. 30 [is shown in] will be described with reference to Fig. 31.

Paragraph beginning at page 53, line 13, has been amended as follows:

The constitution of the auxiliary color creation circuit 2430 shown in Fig. 30 is shown in Fig. 32. In the drawing, numerals 2610 to 2612 indicate subtracters, 2613 to 2615 denote comparators, and 2616 to 2618 denote selectors.

Paragraph beginning at page 54, line 5, has been amended as follows:

A unit for recovering four primary color data which is compressed as a representative color 16 bits in length, an auxiliary color 12 bits in length, and a selection number 4 bits in length as shown in Fig. 29 is constituted by the compression circuit 2220 shown in Fig. 28.

Paragraph beginning at page 54, line 7, has been amended as follows:

The constitution of the [~~compression circuit 2220~~] recovery circuit 2230 is shown in Fig. 33.

Paragraph beginning at page 54, line 10, has been amended as follows:

In Fig. 33, numeral 3210 indicates a compression register, 3220 denotes a recovery processing circuit, and 3230 denotes a recovery register.

Paragraph beginning at page 54, line 24, has been amended as follows:

The constitution of the recovery processing circuit 3220 shown in Fig. 33 [~~is shown in~~] will be described with reference to Fig. 34.

Paragraph beginning at page 55, line 4, has been amended as follows:

The constitution of the processing circuit 3310 shown in Fig. 34 [~~is shown in~~] will be described with reference to Fig. 35.

Paragraph beginning at page 55, line 14, has been amended as follows:

The fourth embodiment of the present invention [~~is~~] has been explained above.

Paragraph beginning at page 56, line 6, has been amended as follows:

In the constitution shown in Fig. 37, comparison data is obtained by calculating the sum of the R, G, and B components as they are. In this constitution, unlike the constitution shown in Fig. 31, the multiplier of each component is not necessary, so that the amount of materials of the hardware can be reduced. The weight of the most significant bit of the G component is [~~the~~] double [~~of~~] that of each of the R and B components. However, when display data is to be read from the frame buffer 220, color data of 4 continuous pixels is compared and the change of each color component is little, so that the comparison data created as mentioned above may be used.

Paragraph beginning at page 56, line 15, has been amended as follows:

In Fig. 38, numerals 2910 to 2912 indicate subtracters and the constitution shown in Fig. 38 is a system [~~that~~] in which an auxiliary color is created by obtaining the difference only from the low-order four bits of the first and second colors. In this constitution, unlike the [~~constitution~~] example shown in Fig. 32, the comparators and selectors are not necessary, so that the amount of [~~materials of the~~] hardware can be reduced. When display data is to be read from the frame buffer 122, the difference is obtained from color data of two pixels among 4 continuous pixels, and the change of each color component is little, and the values of the fifth and sixth bits of each color are equal to each other in 4 continuous pixels and offset by subtraction, so that the first color can be recovered correctly by adding the auxiliary color obtained as mentioned above to the second color (representative color).

Paragraph beginning at page 57, line 11, has been amended as follows:

In the processing circuit 3310 shown in Fig. 40, numerals 3510 to 3512 indicate multipliers, 3513 to 3515 denote selectors, and 3516 to 3518 denote adders.

Paragraph beginning at page 57, line 22, has been amended as follows:

In Fig. 40, the multiplier 3510 outputs a result of doubling the R component of the auxiliary color. The selector 3513 outputs 0 when the selection signal is 0 and outputs the output value of the multiplier 3510 when it is 1. The adder 3516 obtains the sum of the R component of the representative color and the output value of the selector 3513 and outputs it as recovered color data. The adder 3516 outputs

recovered color data also for the B component. The G component is the same as the R component except that the multiplier 2351 outputs [~~the~~] quadruple [~~of~~] the G component of the auxiliary color.

Paragraph beginning at page 58, line 13, has been amended as follows:

In the constitution shown in Fig. 41, comparison data is created by selecting two optional data from the primary color data, and the primary color data having large comparison data is set as a first color, and the other is set as a second color. In this constitution, the image quality is degraded slightly compared with that shown in Fig. 30. However, five comparators and four selectors are not needed in the [~~constitution~~] example shown in Fig. 30, so that the amount of [~~materials of the~~] hardware can be reduced.

Paragraph beginning at page 58, line 28, has been amended as follows:

As shown in Fig. 1 and other drawings, in each of the aforementioned embodiments, the CPU bus 131 has a throughput of 32 bits x 33 MHz and the memory bus 135 has a throughput of 32 bits x 66 MHz. Therefore, even if the CPU 100 accesses the memory 120 up to the limit of the throughput of the CPU bus 131, only half of the throughput of the memory bus 132 is used but the capacity of the memory bus 132 will not be used. Therefore, assuming that the rendering processor 157 does not access the memory 120, a rough estimate of <u>the</u> remainder (66-33) MHz x 32 bits = 1056M bits/second can be used for memory access by the display controller 156. In this case, when a 24-bit full color image of 800 dots x 600 dots is displayed on the CRT 150 at a refresh rate of 60 Hz, the bit rate at which the display

controller 156 is required to read from the memory 120 is (800 x 600) x 60 Hz x 24 bits = 691.2M bits/second. Therefore, even if the CPU 100 accesses the memory 120 up to the limit of the throughput of the CPU bus 131, data is displayed on the CRT 115 without hindrance and the opportunity of access to the memory 120 by the rendering processor 157 can be reserved sufficiently.

Paragraph beginning at page 59, line 20, has been amended as follows:

Actually, it is impossible [~~that~~] for the CPU 100 [~~accesses~~] to access the memory 120 up to the limit of the throughput of the CPU bus 131. Therefore, the throughput of the memory bus 132 may be set to the maximum use rate of the CPU bus 131 and the read rate from the memory 120 which is required by the display controller 156 for display or more.

Paragraph beginning at page 59, line 28, has been amended as follows:

The memory controller explained above has a built-in rendering processor for performing the drawing process. However, the present invention is not limited [~~to it~~] thereto.

Paragraph beginning at page 61, line 25, has been amended as follows:

(1) Coordinate transformation of figure data by CPU 4210 will be explained.

Paragraph beginning at page 62, line 4, has been amended as follows:

(2) Creation of display list by CPU 4210 will be explained.

Paragraph beginning at page 62, line 10, has been amended as follows:

To draw a complicated figure comprising a number of simple figures on the graphics memory 4240, the CPU 4210 converts a drawing command (hereinafter called just a command) into a command format which can be executed by the graphics processor 4220 and transfers it to the graphics memory 4240. Generally, commands in units of a simple figure are combined and connected to commands for one figure. The connected commands are called a display list. The display list is several tens to several hundreds kilobytes in length and is stored in a display list area 4241.

Paragraph beginning at page 62, line 15, has been amended as follows:

(3) Drawing by graphics processor 4220 will be explained.

Paragraph beginning at page 62, line 20, has been amended as follows:

(4) Display by graphics processor 4220 will be explained.

Paragraph beginning at page 63, line 14, has been amended as follows:

The graphics processor 4220 accesses the graphics memory 4240 as mentioned above whenever each element thereof performs a certain process, so that increasing [~~of~~] the access efficiency of the graphics memory 4240 results in improvement of the processing speed. Therefore, the graphics processor 4220 has a cache and FIFO for each access request so as to increase the access efficiency.

Paragraph beginning at pag 63, line 27, has been amended as follows:

A memory controller 4230 receives an access request to the graphics memory 4240 from the caches (1) to (3) and the FIFO 4225, decides on the priority, and controls access. The memory interface circuit 4230 gives priority to access from the display controller 4224. However, while access from the CPU 4210 or the drawing unit 4223 is made, it will not be suspended and access from the display controller is forced to wait.

Paragraph beginning at page 64, line 12, has been amended as follows:

The terminals set the system mode and input [a] clock and reset signals. The graphics processor 4220 can input independent [clocks] clock signals in the drawing and display lines and the drawing line always can perform the high speed process.

Paragraph beginning at page 65, line 22, has been amended as follows:

The drawing commands of the graphics processor are shown in Fig. 44. The Quadrangle Drawing command draws a figure by changing texture data of a rectangle to an optional quadrangle. When the texture data is binary, the command performs color expansion. The LINE command draws a single straight line or a plurality of straight lines. The MOVE command moves the drawing start point. The LOFS command shifts the origin of drawing coordinates. A command after the MOVE command is executed draws a figure by shifting the coordinates of a coordinate parameter shown in the display list by the values designated by this command. The AFFIN command designates rotation, enlargement, or reduction during drawing of a figure. A command rotates (or enlarges or reduces) and draws a

coordinate parameter shown on the display by the value designated by the AFFIN command. The JUMP command branches the display list. The GOSUB command calls the subroutine of the display list. The RET command returns from the subroutine. The TRAP command finishes the display list fetch. The FLASH command makes data existing in the cache (2) 4227 which is a cache of texture data invalid and newly reads data from the graphics memory 4240.

Paragraph beginning at page 68, line 15, has been amended as follows:

Fig. 46 is a [~~function~~] functional block diagram of the CPU FIFO. Whenever the CPU 4210 performs a store operation to the graphics memory 4240, a write request signal is sent from the CPU interface 4221. Then, a counter 4652 is incremented and the write address and data of the CPU 4210 at that time are stored in an FIFO 4650. A match detector 4653 compares the value of the counter 4652 and the FIFO capacity and sets an flip-flop 4658 when it finds that the FIFO is full. As a result, the CPU interface 4221 is notified [~~of~~] that the FIFO 4650 is busy and the CPU 4216 is prevented from storing data any more.

Paragraph beginning at page 69, line 12, has been amended as follows:

When the CPU 4210 does not write data for a given period, a free [~~run~~] running counter 4654 operates so as to write the data of the FIFO 4650 into the graphics memory 4240. When the CPU 4610 reads the graphics memory 4640 or the drawing unit 4223 starts fetching of the display list, the free [~~run~~] running counter 4654 operates so as to write the data of the FIFO 4650 into the graphics memory 4240 prior to them.

Paragraph beginning at page 70, line 10, has been amended as follows:

On the other hand, in the write through system, when the CPU 4210 executes the store instruction for the memory 4240, data is written into the memory 4240 immediately in word units. Therefore, in the FIFO 4650, data to be written in the graphics memory 4240 may be discontinuous addresses. In the worst case, all the addresses may be discontinuous. When the addresses are discontinuous, the writing time into the graphics memory 4240 from the FIFO 4225 of the CPU becomes longer due to the characteristic of the DRAM. In the worst case, the time may be about four times of that of the copy back system (maximum 80 cycles).

Paragraph beginning at page 70, line 23, has been amended as follows:

Next, the constitution and operation of the drawing cache 4228 will be explained. Fig. 47 is a block diagram of the drawing cache. The cache (3) is only for drawing, though the drawing unit 4223 will not read data in the cache (3) 4228. Namely, since the drawing cache 4228 has [~~not a~~] no function for carrying out data operations with a preliminary sketch of the drawing destination, it performs only the writing operation. Since there is no need to read a preliminary sketch, a high speed operation in which the memory access amount is extremely reduced can be performed.

Paragraph beginning at page 71, line 12, has been amended as follows:

A [~~respect~~] common [~~to~~] aspect of the CPU FIFO 4225 and the cache 4228 is that they transfer write data in the number of words detected by the counter 4652 or

the counter 4701 and do not execute useless data transfer. On this point, a general cache used in the CPU or others also transfers data in the portion where no data is rewritten due to writing in line size units.

Paragraph beginning at page 72, line 8, has been amended as follows:

Fig. 49 is a [~~function~~] functional block diagram of the display controller. The display controller 4224 outputs a synchronizing signal (HSYNC, VSYNC) and display data to the display unit 4251 and displays a figure on the screen of the display unit 4251. A timing controller 4946 generates a synchronizing signal (HSYNC, VSYNC) and also [~~notifies~~] indicates the output timing of data in the display buffer 4229 to a display data output controller 4945.

Paragraph beginning at page 73, line 1, has been amended as follows:

The display buffer 4249 reads and writes data as [~~shown~~] indicated below. The display data output controller 4945 reads the display buffer 4229 sequentially from the address shown by a read address register 4942 according to the display dot clock (output of the DCLK terminal of the graphics processor 4220, which is a clock per each pixel of the display unit 4251), outputs the display data to the display unit 4251, and also updates the read address register 4942.

Paragraph beginning at page 75, line 10, has been amended as follows:

Fig. 51 is a time chart showing the operation of an example [~~that~~] in which Ta is 4 times [~~of~~] that shown in Fig. 50. The fixed number B for issuing a display access request is 48 words in length which are 4 times [~~of~~] the number of words of the fixed

number A. The number of continuous words read by one display access also increases and is 64 words in this example. In this connection, display access for one screen in this example is repeated 600 times.

Paragraph beginning at page 76, line 8, has been amended as follows:

[~~Only by~~] By only switching the CAM bit value, the graphics processor can correspond to a different cache system, so that the wide usability of the graphics processor increases. Furthermore, in a system [~~that~~] in which the graphics memory is written by a plurality of CPUs having different cache systems, the display access system of the graphics processor is switched depending on the cache system of a CPU issuing write access, so that graphics by a high speed cache system will not be sacrificed.